COVID-19 data Analysis and Visualization

Md Jisan Uddin Chowdhury (26100)
Rhein-Waal University of Applied Sciences
Md-Jisan-Uddin.Chowdhury@hsrw.org

Statistics
Submitted to Prof. Dr. Frank Zimmer
February 1st, 2021

Contents

Abstract

To clarify the science surrounding COVID-19 to the very wide audience of politicians, scientists, healthcare professionals, and the broader population, visualization strategies have been fully transparent. In this paper, with a help of some small data sets, we have tried to summarize and explain how visualization can help to understand various aspect of the pandemic. This paper provides analysis and visualization of Germany's COVID cases, death events, impact of covid on different age groups of both genders. There is also a brief discussion of presenting data in various ways, how "worldometer" working on covid data, and so on. For the paper, we have used Pandas, Matplotlib, Seaborn, GeoPandas, and NumPy libraries of Python programming language.

Introduction:

Coronavirus was first detected in Wuhan, China, in December 2019. The infection causing virus has been named – "Severe Acute Respiratory Syndrome Coronavirus 2" or "SARS-COV-2". The new name of the disease abbreviated as "COVID-19" stands for, 'CO' for Corona, 'VI' for Virus, and 'D' for Disease. Till today, 96.2 million people have been infected by the virus, where half of the infected people have been recovered and around 2 million people have died, according to the "worldometer.info". So, it is understandable that there have been many different kinds of variation in the dataset of the millions of infected people. The goal of this paper was to work with a simple covid data set with help of different Python libraries just to explore their possibilities.

The Dataset:

We are using several datasets for this project. These datasets contain various information about covid infected people in Germany.

	state	county	age_group	gender	date	cases	deaths	recovered
0	Baden-Württemberg	LK Alb-Donau-Kreis	00-04	F	2020-03-27	1	0	1
1	Baden-Württemberg	LK Alb-Donau-Kreis	00-04	F	2020-03-28	1	0	1
2	Baden-Württemberg	LK Alb-Donau-Kreis	00-04	F	2020-04-03	1	0	1
3	Baden-Württemberg	LK Alb-Donau-Kreis	00-04	F	2020-10-18	1	0	1
4	Baden-Württemberg	LK Alb-Donau-Kreis	00-04	F	2020-10-22	1	0	1

Fig: 1.1 Dataset 1

	ADE	RS	RS_0	GEN	geometry
0	2	02	020000000000	Hamburg	MULTIPOLYGON (((3578695.661 5955304.456, 35781...
1	2	03	030000000000	Niedersachsen	MULTIPOLYGON (((3354775.046 5942939.764, 33546...
2	2	04	040000000000	Bremen	MULTIPOLYGON (((3468658.496 5898364.974, 34702...
3	2	05	050000000000	Nordrhein-Westfalen	POLYGON ((3477450.781 5820982.368, 3479895.578...
4	2	06	060000000000	Hessen	POLYGON ((3535084.230 5721608.644, 3535279.888...

Fig: 1.2 Dataset 2

	state	gender	age_group	population
0	Baden-Wuerttemberg	female	00-04	261674
1	Baden-Wuerttemberg	female	05-14	490822
2	Baden-Wuerttemberg	female	15-34	1293488
3	Baden-Wuerttemberg	female	35-59	1919649
4	Baden-Wuerttemberg	female	60-79	1182736

Fig: 1.3 Dataset 3

The first dataset provides information about the number of infected, dead, and recovered people of both genders in a different month of 2020 in all the states of Germany. There are also geometric data available. The second and third dataset gives us the above-mentioned data. Geometric data will be used for geographic visualization. The figures show the top five rows of each dataset. Among all the first dataset contains 416314 rows and 8 columns. For the purpose of proper analysis and visualization, the values of the different columns have been modified, erased, and changed. The dataset was collected from

various internet-based websites like stack overflow, GitHub and Kaggle.

Differences in Datasets:

Except for Fig 1.2, the datasets that we have mentioned above are almost similar. Our dataset in Fig 1.3 contains important columns of gender, age-group, and population. But it doesn't have the information about infection cases, the number of deaths, and the number of recovered patients. For getting this information we will utilize Fig 1.1. In Fig 1.1 we have got the data of different age-group based on different dates. The whole data set is sorted according to the age-group.

Methods and Materials:

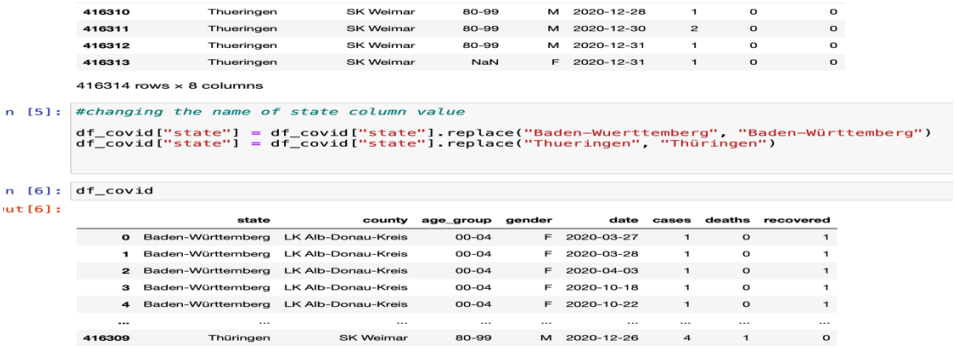
1) Reading Data:

For this project, I had to work with 2 different kinds of file formats. The first one is a .csv (Comma-Separated-Values) and .shp (Shape File Format). A Shapefile format determines a data set's geospatial information as vector properties. These vector properties include points, lines, and polygons. In combination, these attributes can reflect almost any form of shape, such as country borders, spatial points, the flow of rivers, etc. Our purpose for using a shapefile in this project is to represent different states of Germany in a geographic view.

For reading .csv files we have used pandas and for .shp we have used geopandas. Geopandas extends the data types used by pandas to allow spatial operations on geometric types.

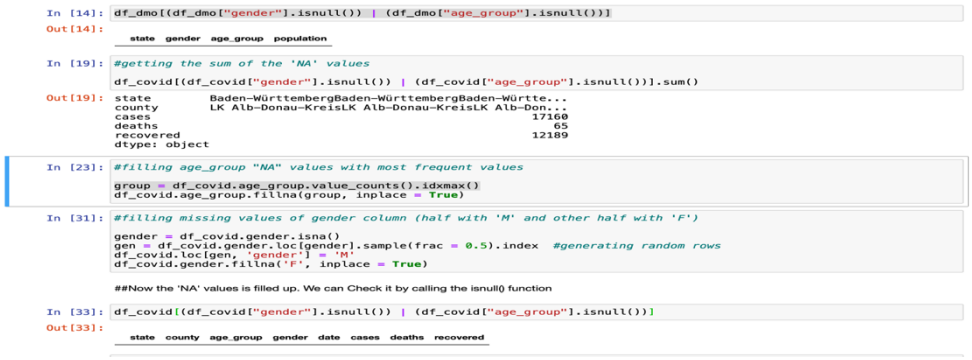
2) Data Manipulating with Pandas:

Manipulation of Data is the modification of information to make it easier to read or more structured. The aim is to properly use the data for analyzing. In our project we have also used some of the basic data manipulation method using the Pandas library such as replacing some values, formatting the "date-time" values, finding the "NA" values, dropping less useful columns.



```
416310  Thueringen  SK Weimar  80-99  M  2020-12-28  1  0  0
416311  Thueringen  SK Weimar  80-99  M  2020-12-30  2  0  0
416312  Thueringen  SK Weimar  80-99  M  2020-12-31  1  0  0
416313  Thueringen  SK Weimar  NaN    F  2020-12-31  1  0  0

416314 rows × 8 columns

In [5]:
#changing the name of state column value
df_covid["state"] = df_covid["state"].replace("Baden-Wuerttemberg", "Baden-Württemberg")
df_covid["state"] = df_covid["state"].replace("Thueringen", "Thüringen")

In [6]:
df_covid

Out[6]:
        state                county          age_group  gender  date        cases  deaths  recovered
0       Baden-Württemberg    LK Alb-Donau-Kreis  00-04  F       2020-03-27  1      0       1
1       Baden-Württemberg    LK Alb-Donau-Kreis  00-04  F       2020-03-28  1      0       1
2       Baden-Württemberg    LK Alb-Donau-Kreis  00-04  F       2020-04-03  1      0       1
3       Baden-Württemberg    LK Alb-Donau-Kreis  00-04  F       2020-10-18  1      0       1
4       Baden-Württemberg    LK Alb-Donau-Kreis  00-04  F       2020-10-22  1      0       1
...     ...                  ...                 ...    ...     ...         ...    ...     ...
416309  Thüringen            SK Weimar           80-99  M       2020-12-26  4      1       0
```

Fig 2: Replacing value names using Pandas



```
In [14]:
df_dmo[(df_dmo["gender"].isnull()) | (df_dmo["age_group"].isnull())]

Out[14]:
  state  gender  age_group  population

In [19]:
#getting the sum of the 'NA' values
df_covid[(df_covid["gender"].isnull()) | (df_covid["age_group"].isnull())].sum()

Out[19]:
state      Baden-WürttembergBaden-WürttembergBaden-Württe...
county     LK Alb-Donau-KreisLK Alb-Donau-KreisLK Alb-Don...
cases                                                  17160
deaths                                                    65
recovered                                              12189
dtype: object

In [23]:
#filling age_group "NA" values with most frequent values
group = df_covid.age_group.value_counts().idxmax()
df_covid.age_group.fillna(group, inplace = True)

In [31]:
#filling missing values of gender column (half with 'M' and other half with 'F')
gender = df_covid.gender.isna()
gen = df_covid.gender.loc[gender].sample(frac = 0.5).index  #generating random rows
df_covid.loc[gen, 'gender'] = 'M'
df_covid.gender.fillna('F', inplace = True)

##Now the 'NA' values is filled up. We can Check it by calling the isnull() function

In [33]:
df_covid[(df_covid["gender"].isnull()) | (df_covid["age_group"].isnull())]

Out[33]:
  state  county  age_group  gender  date  cases  deaths  recovered
```

Fig 3: Filling ‘NA’ values

Filling the ‘NA’ values with most frequently used values, creating random rows in a selected column and so on. We have also done dataset merging. For visualizing our geographical dataset in a convenient way, we have merged 3 datasets into a single dataset and later used it for our geographical data visualization. summarize the data. Since we are working with covid

Data Visualization:

a) <u>Line Graph</u>

Data visualization means representation of data in a graph, chart or other visual format. It actually communicates the data’s connections with images, so that the visual summary of the data can be easier to identify. In this project, we have used several visualization methods to data, so in the very beginning we have used a line plot. Line graphs can be used to evaluate variations over the same period of time with more than one group. It shows data frequencies along a number line.

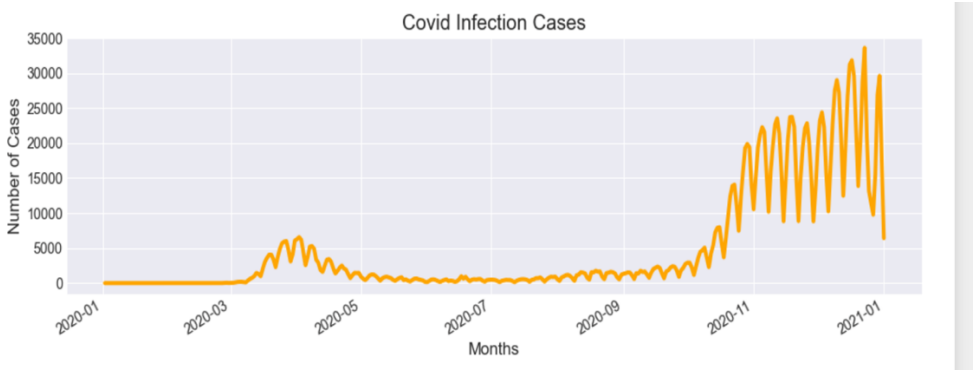


Fig 4: Line graph of covid infection.

For this project, we have simply plotted a line graph to visualize Infection cases in different months. This graph simply shows us the how the number of infections changed by time.

b) Pie Chart

After the line graph, with the help of a pie chart, we have categorized most infected age group in our dataset. Pie charts are used to display percentage or proportional data. The percentage represented by each category is typically given next to the pie slice.

We can see from the chart, by using two columns from our dataset how can we visualize a pie chart. This simple distribution of data increases the chance of sharing insights for everyone involved.

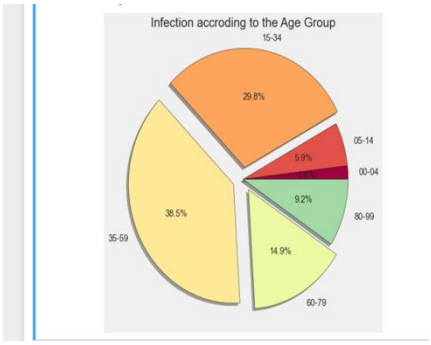


Fig 5: Pie Chart of most infected age groups.

c) Bar Chart:

Then we have used a bar plot to analyze and our dataset. A bar plot illustrates categorical data with rectangular bars with heights or lengths proportional to the values that they represent. Vertically or horizontally, the bars maybe plotted. For our project we have used both types.

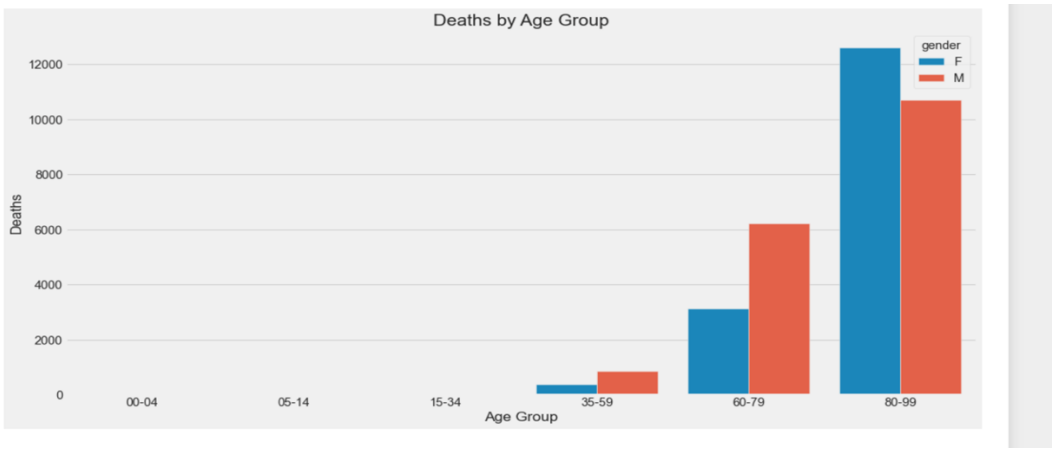


Fig 6: Number of deaths among different gender and age group

With this bar plot we have tried to understand the number of deaths among different age group and gender. We can strongly state that, it has enhanced the ability to retain the attention of the viewer with data that they can understand.

d) <u>Geographical Data:</u>

The last variations of data visualization method we have used for our dataset is Geographical Data. Geodata, also referred to as geographic or geospatial data, refers to data and knowledge specifically or indirectly associated with an Earth-related location. For getting the graph in fig. 7 we had to merge our dataset.

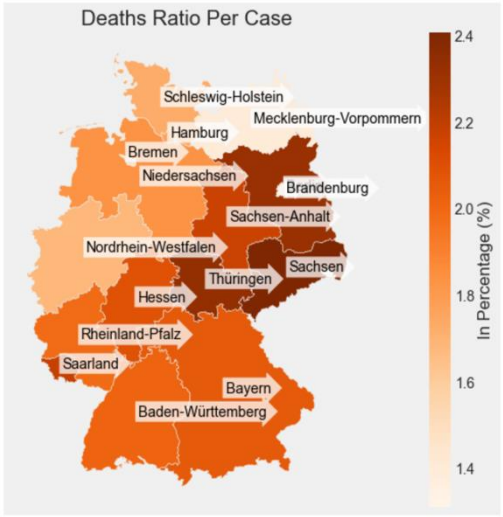


Fig 7: Geographical View of Death Ratio per Case in States

<u>How worldometer works for covid Data:</u>

Worldometer is a website that provides real-time statistics for diverse topics. It is owned and operated by a Chinese

data company Dadax. According to the "worldometer" website, "Wolrdometer manually analyzes, validates, and aggregates data from thousands of sources in real-time and provides global COVID-19 live statistics for a wide audience of caring people around the world".

Sources of collecting data:

Throughout the clock, 24 hours a day, 7 days a week, worlometer captures and processes data. The whole team of analysts and researchers conduct several updates every minute on average, validating the data from an ever-growing list of over 5000 sources under the constant request of users who warn them as soon as an official announcement is made anywhere around the world.

According to the website, the sources include official websites of health ministries or other government agencies and social media pages of government officials. Since national aggregates frequently lag behind the data of the state and local health departments, tracking thousands of regular reports published by local authorities is part of the work of worldometer. A team of a multilingual team also tracks live street press briefings.

Data presenting:

The large amount of data that worldometer collects, is divided into several columns. Users can sort the data they want to see according to the columns. Since they contain information about the covid cases of 219 countries, that means there are 219 rows in the actual dataset and 13 columns. Dataset can be viewed according to ascending or descending order of each column. But by default, the dataset is displayed based on the highest number of total cases. And if there is a big change happen in the "Total Cases" column only then the rows will be sorted automatically. Users can have a look at each country's covid information individually just by clicking on the country name. Total cases and deaths, active cases and deaths, new deaths, and cases, newly infected vs newly recovered, recovery rate vs death rate, etc. can be seen statistically and also with graphical representation. Only for the country USA, the dataset is subdivided into states. Users can have a detailed idea about every state of the USA.

Conclusion:

Working with current pandemic situation is always challenging. The reason behind it is, there are huge amount of data are available and getting a proper and specific dataset sometimes seem very tough. Python is a rich programming language with a bunch of build in libraries.

For this project I have only tried to utilize a very small part of the libraries for analysis and visualization. There are a lot of possibilities of presenting data with the help of the libraries mentioned above. Everything presented and written above was a slight output of the content I have followed through the whole course. I wish and will try to improve the content in the future.

References :

I. https://www.worldometers.info/coronavirus/about/#sources
II.
III. https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/2020-04-29-en.pdf?__blob=publicationFile
IV.
V. https://www.kaggle.com/subinium/simple-matplotlib-visualization-tips#2.-Colormap
VI.
VII. https://icmanaesthesiacovid-19.org/background
VIII.
IX. https://en.wikipedia.org/wiki/Worldometer